ANEXO · TAREAS DEL ALUMNADO

REACTIVIDAD: ESTADO, VALORES DERIVADOS Y EFECTOS

Nombre: _____ Fecha: _____

PRIMERA HORA · PROGRAMA CON EL PROFESOR (40 MIN)

Escribes lo mismo que el profesor en tu `notebox`. Archivo de Figma: 09112026 · Reactivity.

OK	HAZ	ESTÁ HECHO CUANDO...
☐	Cambia `const notes = [` por `let notes = $state([`. Añade la función `addNote()` y el botón **New note**.	Al pulsar el botón aparece una fila nueva y el texto `{notes.length} notes` se actualiza solo.
☐	Añade `let query = $state('');` y un `<input type="search">` con `bind:value={query}`.	El input tiene el estilo del frame 1.
☐	Crea `visible` y `pinnedCount` con `$derived` y usa `visible` en el `{#each}`.	Con `ma` ves el frame 2; con `xyz`, el frame 3.
☐	Añade un `$effect` que pone `NoteBox (n)` en el título de la pestaña.	Al añadir una nota, la pestaña cambia.
☐	Guarda las notas en `localStorage` con otro `$effect` y léelas al empezar (renombra el array a `seed`).	Si la página da error 500, crea `src/routes/+layout.js` con `export const ssr = false;`.
☐	Recarga la página.	Las notas nuevas siguen ahí (DevTools → Application → Local Storage).

Recuerda: si el resultado es un **valor**, es `$derived`. Si el resultado es algo que pasa **fuera** (pestaña, almacenamiento), es `$effect`. Tú cambias el estado; los derivados le siguen. Nunca al revés.

¿QUÉ RUNE? (8 MIN)

En parejas. Para cada valor de una app de compras, escribe **1** = `$state`, **2** = `$derived`, **3** = `$effect` o **puño** = ninguno (una constante o una prop).

EN UNA APP DE COMPRAS...	TU RESPUESTA
Los productos del carrito.	
El precio total.	
El texto escrito en el campo del cupón.	
Si el campo del cupón está vacío.	
Guardar el carrito para que sobreviva a una recarga.	
El nombre de la tienda en la cabecera.	
El número del icono del carrito.	
Enviar «carrito visto» a un servicio de analítica.	

EJERCICIO · SOLO FIJADAS (37 MIN)

Individual, a partir del frame **5 · Exercise** de Figma (los dos estados).

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Añade una casilla «Only pinned» entre la búsqueda y la lista (fila de 44 de alto), con `bind:checked` a un estado nuevo `onlyPinned`.	Al marcarla, la lista solo enseña las notas fijadas.
☐	Usa **un solo** `visible`: la búsqueda **y** la casilla en la misma fórmula.	No hay un segundo derivado ni dos listas.
☐	Muestra el mensaje vacío correcto.	Con la casilla marcada: «No pinned notes match.»; sin marcar, el de antes.
☐	El título de la pestaña dice «NoteBox (1 of 4)». Cambia el efecto que ya tienes; no añadas otro.	La pestaña cambia al buscar y al marcar la casilla.

☐ Si acabas pronto: haz que `query` y `onlyPinned` sobrevivan a una recarga, cada uno con su propia clave. Después, un botón «Clear search» que solo existe cuando `query` no está vacío.

Si algo falla:

- **Has hecho un segundo derivado `visiblePinned` y un `{#if}` que elige entre dos listas.** Funciona, pero piensa en añadir un tercer filtro. Una sola fórmula con `&&` crece; las parejas de listas, no.
- **Tienes un efecto que asigna `visible`.** Es lo que está tachado en la pizarra: usa `$derived`.
- **Al recargar falla `JSON.parse`.** Has guardado `onlyPinned` en la misma clave de `localStorage` que las notas. Una clave para cada cosa.
- **La casilla se marca pero la lista no cambia.** Has escrito `checked={onlyPinned}` sin `bind:`.
- **El título de la pestaña no cambia nunca.** El efecto no lee nada reactivo (una errata o un texto escrito a mano).

ENTREGA

Commit **«Session 3: reactive search»** y push. En Classroom, el enlace y **una frase**: qué valores de tu página son `$state`, cuáles son `$derived` y por qué.

PARA CASA

- Termina el ejercicio.
- En **tu propia app**: haz que tu lista principal esté viva. Al menos dos `$state`, dos `$derived` (uno de ellos, un recuento que se ve en pantalla) y un `$effect`. Haz commit.
- Tutorial: `svelte.dev/tutorial`, parte «Basic Svelte», sección Reactivity.
- Pregunta de transferencia: una app de temporizador muestra minutos y segundos, un botón Start/Pause y suena un aviso al llegar a cero. Escribe su estado, sus valores derivados y sus efectos.